

NO-A143 902

PIPELINED ORTHOGONAL DIGITAL LATTICE FILTERS(U)

1/1

STANFORD UNIV CA INFORMATION SYSTEMS LAB

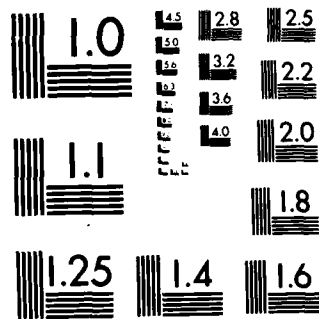
S K RAO ET AL. MAR 84 AFOSR-TR-84-0518 NDA303-79-C-0600

UNCLASSIFIED

F/8 12/1

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

JCALSP  
San Diego

March 1984

A1

14

## PIPELINED ORTHOGONAL DIGITAL LATTICE FILTERS

AUG 5 1984

Sailesh K. Rao and Thomas Kailath  
Information Systems Laboratory  
Stanford University  
Stanford, CA 94305.

## ABSTRACT

An algorithm is presented for the design of systolic arrays that implement single-input single-output time-invariant digital filters. The algorithm is then specialized to the case where the realized array consists only of orthogonal rotational modules and delay elements interconnected in such a manner as to render the circuit pipelineable.

## 1. INTRODUCTION

Consider a single-input single-output linear time-invariant discrete-time system that allows a transfer function description:

$$H(z) = \frac{\sum_{i=0}^n N_i z^{-i}}{z^n + \sum_{i=1}^n D_i z^{-i}} = \frac{N(z)}{D(z)} \quad (1)$$

The system is said to be strictly stable if the function,  $H(z)$ , in the complex variable,  $z$ , is analytic on the unit circle,  $\Pi$ , and in the region,  $E$ , where,  $|z| > 1$ . In other words, the zeros of the polynomial,  $D(z)$  should lie in the region  $D$ , where,  $|z| < 1$ . A strictly stable function is said to be a Schur function if it has magnitude less than or equal to unity on  $\Pi$ .

In this paper, the transpose of a matrix is denoted by the superscript  $T$ , the complex conjugate transpose by an over-bar, and the para-Hermitian conjugate by the subscript  $*$ . The para-Hermitian conjugate of a matrix function is, of course, defined as:

$$A_*(z) = \bar{A}(1/z)$$

Given a transfer function,  $H(z)$ , the synthesis problem is to obtain a hardware implementation of  $H(z)$ . The quality of such an implementation using VLSI technology is then dependent upon the following criteria:

- (1) **Modularity:** The circuit should preferably be a regular interconnection of similar processing elements.
- (2) **Pipelineability:** The maximum throughput achievable should be independent of the number of processing elements,  $n$ , in the array.
- (3) **Uniformity:** It is desirable to design VLSI circuits that can implement digital filters with a variety of transfer functions of different orders by merely changing certain parameters of the circuit.

This work was supported in part by the Defense Research Projects Agency under Contract MDA903-79-C-0980, by the U.S. Army Research Office, under Contract DAAG29-79-C-0215, and by the Air Force Office of Scientific Research, Air Force Systems Command under Contract AF49-620-79-C-0058.

- (4) **Communication:** Communication between the processing elements should be restricted to immediate neighbours. A circuit that is modular and has only nearest-neighbour links is said to be systolic [2].
- (5) **Numerical Stability:** There are certain undesirable phenomena, e.g., limit-cycle and overflow oscillations, associated with the finite-precision implementation of linear systems that have been well-studied in the literature [3]. The circuit should be free of them. It should also be insensitive to slight variations in the multiplicative parameters in the implementation.

In the next section, a systolic array configuration is proposed for the circuit that, in general, satisfies properties (1), (3), (4). The pipelineability of such an array is considered and it will be shown that every strictly stable transfer function can be realized in several different ways that ensure that all the above desirable criteria are met.

## 2. BI-DIRECTIONAL LINEAR SYSTOLIC ARRAYS

A systolic array is said to be *bi-directional* if every processing element receives input from and provides output to each of its neighbours. (The external environment is a neighbour to the processing elements at the corners of the array.) The array is said to be *linear in operation* if each processing element performs only linear operations on its inputs. It is said to be *linear in configuration* if each processing element has only two neighbours. A systolic array that is linear both in operation and in configuration is *linear*. Such an array is *time-invariant* if the  $z$ -transform of the vector output sequence,  $Y_i(z)$ , of its  $i^{\text{th}}$  processing element is related to the  $z$ -transform of its corresponding vector input sequence,  $U_i(z)$ , by:

$$Y_i(z) = \Sigma_i(z) U_i(z) \quad (2)$$

where  $\Sigma_i(z)$  is some rational matrix function of  $z$ , necessarily square, such that  $\Sigma_i(\infty)$  has only finite elements, in order to ensure causality. A linear systolic array is said to have a *ternary signature* if it has either (a) two wires flowing from right to left and one from left to right, or (b) vice-versa. (See Fig. 1)

The intent of this paper is to present a general framework for designing such arrays so that when they are suitably terminated at both ends with constant multiplier interconnections, they are realizations for  $H(z)$ . The reasons for choosing ternary signature arrays as opposed to binary signature arrays are the following:

A pipelined orthogonal realization of  $H(z)$  in a binary signature array requires, in general, at least  $2n$  delay elements and  $(3n + 1)$  elementary Givens rotational

Approved for public release;  
distribution unlimited.

Grant - AFOSR-83-C-228

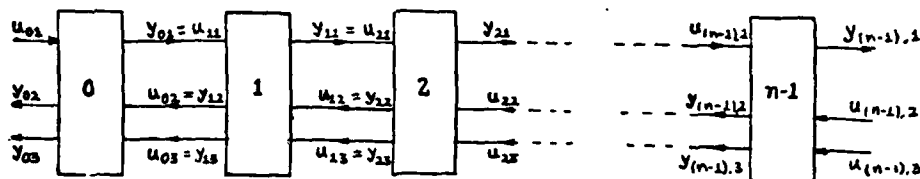


FIG. 1

modules for its implementation. (See [1]). On the other hand, ternary signature arrays with  $n$  delay elements and  $(2n + 1)$  elementary Givens rotational modules will be derived in what follows. Another reason is that even if we synthesize the filter in a binary signature array with the attendant increase in hardware requirement, the speed at which the array can process the input sequence is only half the speed that is achievable using ternary signature arrays. Clearly, there is no tradeoff here.

Consider a linear time-invariant systolic array with a ternary signature as shown in Fig. 1. Suppose that for some unknown constant termination at one end of the array, say the right end, the input to output transfer at the left end is given by: (with reference to Fig. 1.)

$$\begin{bmatrix} y_{02}(z) \\ y_{03}(z) \end{bmatrix} = \frac{1}{Q(z)} \begin{bmatrix} P(z) \\ R(z) \end{bmatrix} u_{01}(z) = \varphi(z) u_{01}(z) \quad (3)$$

The function,  $\varphi(z)$ , will be referred to as the *generating function* of the systolic array, for reasons that will become clear in what follows. The *degree* of the function,  $\varphi(z)$ , is the polynomial-degree of  $Q(z)$ , assuming that the greatest common divisor of  $Q(z)$ ,  $P(z)$ ,  $R(z)$  is constant.

For a given transfer function,  $H(z)$ , there are several possible choices of  $\varphi(z)$ . Suppose that the designer wishes to use

$$u(z) = k_1 u_{01}(z) + k_2 y_{02}(z) \quad (4a)$$

$$y(z) = y_{03}(z) \quad (4b)$$

such that the transfer function from  $u(z)$  to  $y(z)$  is  $H(z)$ . Then,  $\varphi(z)$  should be such that  $P(z) \cdot Q(z) \cdot R(z)$  are polynomials satisfying the relationship:

$$R(z) = N(z) \quad (5a)$$

$$k_1 Q(z) + k_2 P(z) = D(z) \quad (5b)$$

Clearly, the choice of  $P(z)$  and  $Q(z)$  is not unique. Moreover, the designer might wish to use different relationships in eqn. (4) in order to obtain the desired input and output terminals and the nonuniqueness is thus further enhanced.

We will get back to this problem of the choice of the generating function later. For the present, assume that  $\varphi(z)$  is known.

The realization procedure, then, depends upon the solution to the following problem:

Given the  $(2 \times 1)$  vector generating function,  $\varphi(z)$ , of degree,  $n$ , obtain a linear time-invariant systolic array with  $n$  processing elements and the constant termination at the right end of the array in Fig. 1, such that the input to output transfer at the left

end of the array is  $\varphi(z)$ .

The following algorithm solves the above problem:

#### The Systolic Realization Algorithm:

It is required that the systolic array in Fig. 1 with the  $i^{\text{th}}$  processing element defined by:

$$\begin{bmatrix} y_{i1} & y_{i2} & y_{i3} \end{bmatrix}^T = \Sigma_i(z) \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} \end{bmatrix}^T \quad (6)$$

is such that when the right end of the array is terminated with

$$\begin{bmatrix} u_{(n-1)2} & u_{(n-1)3} \end{bmatrix}^T = \varphi_n(z) y_{(n-1)1} \quad (7)$$

the signals at the left end of the array are related by:

$$\begin{bmatrix} y_{02} & y_{03} \end{bmatrix}^T = \varphi(z) u_{01}(z) \quad (8)$$

and that  $\varphi_n(z)$  in eqn. (7) is a constant vector.

**Step 1: Initialization:** Let  $\varphi_0(z) = \varphi(z)$ .

For  $i = 0, 1, \dots, n-1$ , do

**Step 2:** Choose the complex scalar constants,  $\alpha_i, \beta_i$ , the  $(2 \times 2)$  nonsingular constant matrix,  $C_i$ , and the  $(1 \times 2)$  constant vector,  $v_i^T$ , and obtain (we will restrict these choices later in order to ensure that the array is pipelineable and orthogonal. For the present, assume that they are free parameters subject to the designer's fancy and the equations below.)

$$\varphi_{i+1}(z) = \frac{C_i [\varphi_i(z) - \varphi_i(\beta_i)]}{v_i^T [\varphi_i(z) - \varphi_i(\alpha_i)]} p_i(z) \quad (9a)$$

such that the  $(3 \times 3)$  constant matrix,  $\Theta_i$ , defined by:

$$\Theta_i = \begin{bmatrix} -v_i^T \varphi_i(\alpha_i) & v_i^T \\ -C_i \varphi_i(\beta_i) & C_i \end{bmatrix} \quad (9b)$$

is nonsingular,

$$\alpha_i \neq \beta_i \quad (9c)$$

and

$$\varphi_i(\alpha_i), \varphi_i(\beta_i) \neq \infty \quad (9d)$$

and  $p_i(z)$  is chosen to be:

$$p_i(z) = \frac{z - \alpha_i}{z - \beta_i}, \quad \alpha_i, \beta_i \neq \infty \quad (9e)$$

$$p_i(z) = z - \alpha_i, \quad \beta_i = \infty, \alpha_i \neq \infty \quad (9f)$$

**Step 3: Form**

$$\Sigma_i(z) = \begin{bmatrix} p_i^{-1}(z) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -v_i^T (\varphi_i(\alpha_i) - \varphi_i(\beta_i)) & v_i^T C_i^{-1} \\ \varphi_i(\beta_i) & C_i^{-1} \end{bmatrix} \quad (10)$$

end.

**Comments:** The constraint (9d) imposed in step 2 can be relaxed in certain cases. Details can be found in [1]. Constraint (9b) is necessary as shown below, and (9c) is required to ensure causality.

**Proof of the algorithm:** Using eqn.(10) in eqn.(6), it can be shown, after some simple algebra, that

$$\begin{bmatrix} y_{i1}(z) \\ y_{i2}(z) \\ y_{i3}(z) \end{bmatrix} = \begin{bmatrix} p_i^{-1}(z) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Theta_i \begin{bmatrix} u_{i1}(z) \\ y_{i2}(z) \\ y_{i3}(z) \end{bmatrix} \quad (11)$$

Now, if,

$$\begin{bmatrix} y_{i2}(z) \\ y_{i3}(z) \end{bmatrix} = \varphi_i(z) u_{i1}(z) \quad (12)$$

i.e.,  $\varphi_i(z)$  is the generating function for the  $i^{\text{th}}$  sub-array that is the cascade of processing elements that are indexed  $i$  or greater in Fig. 1., then, by substituting eqn.(12) in eqn.(11), using the defining relationship in eqn.(9a) and the interconnecting equations displayed in Fig. 1, we have  $\varphi_{i+1}(z)$  is the generating function for the  $(i+1)^{\text{th}}$  sub-array.

The assumption that  $\Theta_i$  is invertible ensures that the converse of this statement is true, i.e., if  $\varphi_{i+1}(z)$  is the generating function for the  $(i+1)^{\text{th}}$  sub-array and if the  $i^{\text{th}}$  processing element has a transfer defined by eqns(6,10), then  $\varphi_i(z)$  is the generating function for the  $i^{\text{th}}$  sub-array. (The invertibility of  $\Theta_i$  is necessary in order to state this [1].) Hence, if the right end of the array in Fig.1 is terminated according to eqn.(7), then the input to output transfer function at the left end of the array is  $\varphi(z)$  as desired.

It is easy to check that if  $\varphi_i(z)$  is of degree,  $n_i$ , then,  $\varphi_{i+1}(z)$  is of degree  $n_i - 1$  at most. It has been proved in [1] that the degree is exactly  $n_i - 1$ . Hence,  $\varphi_n(z)$  is a constant vector.

**Comments:** A remarkable property of the systolic realization algorithm is that almost every known canonical realization of digital filters, e.g., controller, controllability, cascade, parallel [5], all the different versions of the Gray-Markel lattice filter[6], wave-digital translations of passive analog Foster and Cauer canonical forms [7], the orthogonal digital filter realizations of Deprettere et al[8], to name a few, can be obtained by certain choices of the various parameters used in the algorithm. Moreover, for every realization obtained using the systolic realization algorithm, there is a dual realization that can be obtained using the duality theorem in [1], e.g., the observer form realization is dual to the controller form realization, etc.

#### Properties of The Systolic Realization Algorithm:

Some relevant properties of the systolic realization algorithm are given below. A detailed discussion of these and other properties together with their proofs can be found in [1].

(1) The systolic array obtained by the algorithm is in general a *non-computable* cascade. However, it can always be made computable by executing bilinear transformations on the functions,  $\varphi_i(z)$ , at each step of the algorithm.

(2) The array leads to a computable cascade as it is if

$$\alpha_i = 0; \quad \beta_i = \infty, \quad \text{for all } i \quad (13)$$

This results in  $p_i(z) = z$  for all  $i$ .

(3) **Pipelineability:** The array leads to a pipelineable implementation if  $\alpha_i, \beta_i$  are chosen as in eqn.(13) [1,9]. The throughput is maximum, all other factors being the same, *if and only if* this choice is adhered to[1]. Such arrays will be referred to as *completely pipelineable arrays* in what follows.

#### 3. COMPLETELY PIPELINEABLE ORTHOGONAL ARRAYS

The systolic array in Fig.1 is said to be *orthogonal* if every processing element can be implemented as an interconnection of orthogonal rotational modules and  $z^{-1}$  blocks. If, in addition, the array is completely pipelineable, it is a *completely pipelineable, orthogonal array*. It has been shown in [1] that such arrays have very good numerical properties and therefore, satisfy all the desirable criteria mentioned above for VLSI implementation.

A necessary and sufficient condition for a completely pipelineable array to be orthogonal is that the constant matrix,  $\Gamma_i$ , defined by:

$$\Sigma_i(z) = \text{diag}[z^{-1}, 1, 1] \Gamma_i \quad (14)$$

is an orthogonal matrix for all  $i$ . Using the definition of orthogonality and eqn. (10), this reduces to the conditions:

$$\bar{C}_i C_i = [I - \varphi_i(\infty) \bar{\varphi}_i(\infty)]^{-1} \quad (15a)$$

$$\bar{\varphi}_i^T \varphi_i^T = \varphi_i(\infty) \bar{\varphi}_i(\infty) (1 - \bar{\varphi}_i(\infty) \varphi_i(\infty))^{-1} \quad (15b)$$

$$\varphi_i(0) \bar{\varphi}_i(\infty) = 1 \quad (15c)$$

Eqn. (15) is solvable if and only if  $|\varphi_i(\infty)| < 1$  and eqn.(15c) is true for  $\varphi_i(z)$ . The systolic realization algorithm then reduces to the Schur algorithm[10].

The question that now requires to be addressed is the following: *For what choices of the generating function, can the above equations be solved for all  $i$ ?* Theorem 1, stated below, a special case of which is originally due to Schur[10], clarifies the situation:

**Theorem 1:** A completely pipelineable orthogonal array exists for any generating function,  $\varphi(z)$ , *if and only if:*

[a]  $\varphi(z)$  is composed of scalar rational Schur functions and

[b]  $\varphi(z)$  satisfies the equation:

$$1 - \varphi(z) \varphi(z) = \frac{\sigma^2}{Q(z) \bar{Q}(z)} \quad (16)$$

where  $\sigma$  is some real constant, and  $Q(z)$  is the polynomial defined in eqn.(3).

In addition, if in eqn.(16),  $\sigma$  is nonzero, then the termination,  $\varphi_n$ , is such that  $\bar{\varphi}_n \varphi_n < 1$ . Else,  $\bar{\varphi}_n \varphi_n = 1$ .

The proof for the above theorem can be found in [1], wherein a more general theorem is stated that defines the conditions for the existence of orthogonal arrays as such.

#### Choice of the generating function:

Given a  $(2 \times 1)$  function,  $\varphi(z)$ , that satisfies eqn.(16), and the restrictive choices of the parameters,  $C_i, \alpha_i, \beta_i, \gamma_i$ , as defined in eqn.(15), the systolic realization algorithm yields a completely pipelineable orthogonal array. The next step is to obtain this generating function  $\varphi(z)$ , from the given scalar transfer function,  $H(z)$ . Again, there are several ways in which this choice can be made.

(1) First, the terminating conditions at the left end of the array have to be chosen.

(2) Secondly, the constant  $\sigma$  in eqn.(16) has to be chosen.

The restrictions for making these choices and the general properties of the resulting arrays can be found in [1]. For the present, suppose the terminating conditions are given as in eqn.(4.5) and that the constant  $\sigma = 0$  in eqn.(16). Even then, there are two constant parameters,  $k_1, k_2$  available to the designer, and only two particular choices of these parameters are considered below:

(1) *The Direct Embedding:*

Suppose  $k_2 = 0$ . Then,  $P(z), Q(z), R(z)$  have to satisfy the relationships:

$$R(z) = N(z) \quad (17a)$$

$$Q(z) = D(z)/k_1 \quad (17b)$$

$$P(z)P_0(z) = D(z)D_0(z)/k_1^2 - N(z)N_0(z) \quad (17c)$$

From the well-known spectral factorization theorem, it can be derived that eqn.(17c) is solvable if and only if

$$|k_1| \leq \frac{1}{|H(e^{jw})|}, \quad \text{for all } 0 \leq w \leq 2\pi \quad (17d)$$

The resulting pipelined orthogonal array together with some simulation results have been discussed in depth in [11], wherein it has also been shown that the solution to eqn.(17c) is trivially obtainable when  $H(z)$  is a Butterworth, or a Tchebycheff, or an Elliptic function, for  $k_1 = 1$ .

(2) *The Darlington Embedding:*

Consider the choice,  $k_1 = k_2 = 1$ . Then,

$$R(z) = N(z) \quad (18a)$$

$$Q(z) + P(z) = D(z) \quad (18b)$$

then, if  $Q(z), P(z)$  are written as:

$$Q(z) = (D(z) + C(z))/2 \quad (18c)$$

$$P(z) = (D(z) - C(z))/2 \quad (18d)$$

then

$$C(z)D_0(z) + C_0(z)D(z) = 2N(z)N_0(z) \quad (18e)$$

Eqn.(18) is precisely the one used by Deprettere et al[8] for obtaining the generating function for their binary signature orthogonal array. The systolic realization algorithm is then analogous to the Darlington synthesis procedure for passive analog filters.

For this termination, however, the designer has to be careful to ensure that the resulting structure is computable, by scaling the transfer function such that  $R(\infty)/Q(\infty) = 0$ .

A general methodology for obtaining the generating function so that the resulting array is orthogonal is available in [1].

**Discussion:** In eqn.(15), the matrix,  $C_i$  can be chosen to be lower or upper triangular, in which case, the processing element can be implemented using two elementary Givens rotational modules[1].

In [1], a processor configuration termed the **Universal Schur Filter** has been proposed based upon the results presented here, that is capable of implementing time-invariant systems of arbitrary order, if the processing elements have sufficient memory. The arithmetic unit of the processing element implements the CORDIC algorithm[4] and the logical unit consists of a few counters and gates. The operation of the circuit is assumed to be asynchronous, with the necessary handshaking provided using binary semaphores. A description of the processing element in Concurrent Pascal is given. A VLSI circuit that has at least one such processing element and has

access to sufficient memory is capable of implementing any strictly stable linear system in such a manner that the circuit is free of limit-cycle and overflow oscillations and has low coefficient sensitivity. The throughput of the circuit is then directly proportional to the number of processing elements in the circuit, provided this number is an integer submultiple of  $n$ . The throughput is not influenced by having more than  $n$  processing elements in the circuit. The circuit is also capable of implementing vector transfer functions of arbitrary order.

**CONCLUSIONS**

In this paper, the systolic realization algorithm for synthesizing digital filters on a VLSI circuit has been presented. The problem of implementing digital filters on a systolic array in a pipelineable fashion has been solved in a numerically robust manner. For a more complete treatment of the VLSI synthesis of digital filters, the interested reader is referred to [1].

**REFERENCES**

- [1] Sailesh K.Rao and Thomas Kailath, 'Digital Filtering in VLSI', Technical Rept., Information Systems Lab, Dept. EE, Stanford Univ., Jan. 1984.
- [2] H.T.Kung, 'Why Systolic Architectures?', IEEE Computer, vol. 25, No.1, pp37-46, Jan.1982.
- [3] L.R.Rabiner and B.Gold, *Theory and Applications of Digital Signal Processing*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1975.
- [4] J.E.Volder, 'The CORDIC Trigonometric Computing Technique,' IRE Trans. on Electronic Computers, vol. EC-8, pp330-334, Sep. 1959.
- [5] T.Kailath, *Linear Systems*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1980.
- [6] A.H.Gray, Jr., and J.D.Markel, 'A Normalized Digital Filter Structure,' IEEE Trans Acoust., Speech, Signal Processing, vol.ASSP-23, pp.486-494, Oct.1975.
- [7] A.Fettweis, 'Digital Filter Structures Related to Classical Filter Networks,' A.E.U. band 25, pp.79-89, 1971.
- [8] E.Deprettere and P.Dewilde, *Orthogonal Cascade Realization of Real Multiport Digital Filters*, Tech. Rep., Network Theory Section, Dept. of Elec. Engg., Delft Univ. of Tech., Delft, The Netherlands, 1980.
- [9] H.V.Jagadish, T.Kailath, J.A.Newkirk, and R.G.Mathews, 'Pipelining in Systolic Arrays,' Conf. Rec. Seventeenth Asilomar Conference on Circuits and Systems, Pacific Grove, 1983.
- [10] I.Schur, 'Ueber Potenzreihen, die im Innern des Einheitskreises beschränkt sind,' J. für die Reine und Angewandte Mathematik, vol. 147, pp.205-232, Berlin, 1917.
- [11] Sailesh K.Rao and Thomas Kailath, 'Orthogonal Digital Filters for VLSI Implementation', to appear in the IEEE Trans. on Circuits Syst.



Handwritten signature or initials.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS													
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.													
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE															
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 84-0518													
6a. NAME OF PERFORMING ORGANIZATION Stanford University	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research													
6c. ADDRESS (City, State and ZIP Code) Department of Electrical Engineering Stanford CA 94305		7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332													
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-83-0228													
8c. ADDRESS (City, State and ZIP Code) Bolling AFB DC 20332		10. SOURCE OF FUNDING NOS. <table border="1"><thead><tr><th>PROGRAM ELEMENT NO.</th><th>PROJECT NO.</th><th>TASK NO.</th><th>WORK UNIT NO.</th></tr></thead><tbody><tr><td>61102F</td><td>2304</td><td>A6</td><td></td></tr></tbody></table>		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.	61102F	2304	A6					
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.												
61102F	2304	A6													
1. TITLE (Include Security Classification) PIPELINED ORTHOGONAL DIGITAL LATTICE FILTERS.															
2. PERSONAL AUTHOR(S) S.K. Rao and T. Kailath															
13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) MAR 1984	15. PAGE COUNT 4												
5. SUPPLEMENTARY NOTATION ICASSP, MAR 1984, San Diego, California.															
7. COSATI CODES <table border="1"><thead><tr><th>FIELD</th><th>GROUP</th><th>SUB. GR.</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></tbody></table>		FIELD	GROUP	SUB. GR.										18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.													
9. ABSTRACT (Continue on reverse if necessary and identify by block number) An algorithm is presented for the design of systolic arrays that implement single-input single-output time-invariant digital filters. The algorithm is then specialized to the case where the realized array consists only of orthogonal rotational modules and delay elements interconnected in such a manner as to render the circuit pipelineable.															
DISTRIBUTION/AVAILABILITY OF ABSTRACT CLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED													
22. NAME OF RESPONSIBLE INDIVIDUAL CPT Brian W. Woodruff		22b. TELEPHONE NUMBER (Include Area Code) (702) 767- 5017	22c. OFFICE SYMBOL NM												

END

FILMED

9-64

ERIC